

Counting Integer Points in Parametric Polytopes Using Barvinok’s Rational Functions¹

Sven Verdoolaege,² Rachid Seghir,³ Kristof Beyls,⁴ Vincent Loechner,³
and Maurice Bruynooghe²

Abstract. Many compiler optimization techniques depend on the ability to calculate the number of elements that satisfy certain conditions. If these conditions can be represented by linear constraints, then such problems are equivalent to counting the number of integer points in (possibly) parametric polytopes.

It is well known that the enumerator of such a set can be represented by an explicit function consisting of a set of quasi-polynomials, each associated with a chamber in the parameter space. Previously, interpolation was used to obtain these quasi-polynomials, but this technique has several disadvantages. Its worst-case computation time for a single quasi-polynomial is exponential in the input size, even for fixed dimensions. The worst-case size of such a quasi-polynomial (measured in bits needed to represent the quasi-polynomial) is also exponential in the input size. Under certain conditions this technique even fails to produce a solution.

Our main contribution is a novel method for calculating the required quasi-polynomials *analytically*. It extends an existing method, based on Barvinok’s decomposition, for counting the number of integer points in a non-parametric polytope. Our technique always produces a solution and computes polynomially-sized enumerators in polynomial time (for fixed dimensions).

Key Words. Barvinok’s decomposition, Compiler analysis, Ehrhart quasi-polynomial, Parametric polytope, Polyhedral model, Vector partition function, Signed unimodular decomposition.

1. Introduction. In many program analyses and optimizations, questions starting with “how many” need to be answered, e.g.,

- How many memory locations are touched by a loop? [23]
- How many operations are performed by a loop? [35]
- How many cache lines are touched by a loop? [23]
- How many array elements are accessed between two points in time? [7]
- How many array elements are live at a given iteration (i, j) ? [1], [2], [34], [52]
- How many times is a statement executed before an iteration (i, j) ? [48], [25]
- How many parallel processing elements can be used when executing a loop on an FPGA? [6], [19], [24], [29]
- How many cache misses does a loop generate? [15], [13], [27]

¹ Kristof Beyls was supported by research projects GOA-12051002 and IWT-SB991147. Sven Verdoolaege was supported by FWO-Vlaanderen.

² Department of Computer Science, Katholieke Universiteit Leuven, 3000 Leuven, Belgium. {sven,maurice}@cs.kuleuven.be.

³ ICPS-LSIIT UMR 7005, Université Louis Pasteur, F-67070 Strasbourg, France. {seghir,loechner}@icps.u-strasbg.fr.

⁴ Department of Electronics and Information Systems, Ghent University, Ghent, Belgium. kristof.beyls@elis.UGent.be.

The remainder of this paper is not included as this paper is copyrighted material. If you wish to obtain an electronic version of this paper, please send an email to bib@elis.UGent.be with a request for publication P107.205.pdf.
